AD-A101 680 RESEARCH TRIANGLE INST RESEARCH TRIANGLE PARK NC SYST--ETC F/G 9/2
DESIGNING LINEAR STORAGE HIERARCHIES SO AS TO MAXIMIZE RELIABIL--ETC(U)
SEP 79 K S TRIVEDI N00014-79-C-0571
UNCLASSIFIED RTI/1840/00-01F NL

1 OF 1
AD
A101680

END
DATE
FILMED
8-81
DTIC

RESEARCH TRIANGLE INSTITUTE

14 RTI/1840/00-01 F

11 September 1979

LEVEL

AD A101680

6 DESIGNING LINEAR STORAGE HIERARCHIES SO AS TO MAXIMIZE RELIABILITY SUBJECT TO COST AND PERFORMANCE CONSTRAINTS

12

15 Contract No. N00014-79-C-0571

DTIC

21 1981

G

9 Final Report

10 K. S. /Tri

Prepared for

Office of Naval Research
Washington, DC

Prepared by

Systems and Measurements Division
Research Triangle Institute
Research Triangle Park, NC 27709

and

Computer Science Department
Duke University
Durham, NC 27706

41 034

RESEARCH TRIANGLE PARK, NORTH CAROLINA 27709

81 7 17 001

(2) cab
10/79

RESEARCH TRIANGLE INSTITUTE
POST OFFICE BOX 12194
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27709

SYSTEMS AND MEASUREMENTS DIVISION

October 18, 1979

Dr. John Dimmock
Office of Naval Research
Code 427
800 North Quincey Street
Arlington, VA 22217

Dear John:

Attached is a copy of our final report entitled "Designing Linear Storage Hierarchies so as to Maximize Reliability Subject to Cost and Performance Constraints." With your permission, we would like to submit this paper to the Seventh International Symposium on Computer Architecture to be held in May, 1980. This work was performed under your Office of Naval Research Contract No. N00014-79-C-0571 with the Research Triangle Institute.

We believe that this report presents an original view of how digital systems' reliability, performance and cost can be considered in a unified manner.

In the course of this study, we identified several areas which we believe deserve further consideration. For example, in the present work fault-tolerance concepts are not totally addressed - only a simplex system is considered. We believe that in a VHSIC environment device reliability and its enhancement through fault-tolerance techniques should be considered more carefully. Our future proposed work will involve extending the present results to the design of systems with fault tolerance. Specifically, CPU subsystems using hybrid N-modular redundancy configurations will be considered.

I am looking forward to seeing you in the near future. Thank you once again for supporting this work.

Sincerely yours,

Jim

James B. Clary, Supervisor
Digital Systems

JBC/cn

Attachment

Designing Linear Storage Hierarchies so as to Maximize Reliability Subject to Cost and Performance Constraints*

K.S. Trivedi

Department of Computer Science
Duke University
Durham, N.C. 27706

August 1979

* This work was supported in part by the Office of Naval Rsearch under Contract No. N00014-79-C-0571.

Abstract

A geometric programming model is proposed to determine the optimal design of the CPU and its matching storage hierarchy. The objective function is the maximization of system reliability subject to performance and budgetary limitations. Examples illustrating the use of the model are presented.

1. Introduction

An important area of concern during the design of a computer system is the design of its storage subsystem. Several optimization models for the design of storage hierarchies are available [3,4,8,12]. These models typically optimize system performance, measured by throughput or average hierarchy access time, subject to budgetary limitations. In the previous efforts, no consideration is given to the reliability issues. In systems designed for avionics, space, and certain military applications, reliability is of utmost concern. In this paper, we develop an optimization model for linear storage hierarchies with the objective of maximizing system reliability subject to a cost and a performance constraint. The capacities of various memory levels are the decision variables. An extended model also includes the CPU speed as a decision variable.

Chow [3,4], considered the design of a linear storage hierarchy with the objective of minimizing the average access time subject to a cost constraint. The present paper is an extension of Chow's model to include reliability considerations. Other related work is by MacDonald and Sigworth [8], and Trivedi and Sigmon [12].

2. Model Development

The storage hierarchy consists of n levels, $M_1, M_2, \ldots, M_n$, as shown in Figure 1. The hierarchy management follows the staging rules proposed by Gecsei [6]. The execution is out of the fastest level M, which may be thought of as a cache. Information transfers are between adjacent levels only. By convention, the higher the level (smaller the index) in the hierarchy, the smaller its capacity. Rules of operation of such a hierarchy are described in [6,10].

The capacity of memory level M_i, will be denoted by K_i (bits), and the block size by s_i (bits). Since the entire address space of the program must be contained in level n, we assume that the capacity of that level, K_n, is a fixed input parameter to the model. The decision variables are the capacities $K_1, K_2, \ldots, K_{n-1}$.

The behavior of the program can be characterized by a reference string. We assume, however, that the program behavior and the effect of storage management strategy are compacted into a single function called the success function which gives the probability that a storage reference from the CPU is found in a given level. The success function H_i, for level i depends on the capacity of level i (K_i), the block size of level i and all of the lower levels ($s_i, s_{i+1}, \ldots, s_n$), and the block replacement algorithm. If the hierarchy management rules satisfy certain reasonable properties [6], it can be shown that the success function, H_i, depends only upon the capacity K_i and the block size s_i. That is,

$$H_i = H_i(K_i; s_i) \qquad i=1,2,\ldots,n \quad .$$

The semicolon is used above to emphasize the fact that K_i is a decision variable while s_i is a fixed parameter. The miss ratio function is defined to be

$$F_i = 1 - H_i \quad .$$

Let t_i be the average time to transfer a block of size s_{i-1} from the i^{th} level to level i-1. The access times $t_1, \ldots, t_n$ are considered fixed parameters in our model. Due to the linear organization, the time to fetch a block from level i and percolate it (or appropriate sub-blocks) up to level 1, denoted by T_i, is given by

$$T_i = \sum_{j=1}^{i} t_j \quad .$$

The average access time, T, of the entire hierarchy is the weighted average of the access time to various levels, that is,

$$T = \sum_{i=1}^{n} h_i T_i \quad .$$

The weight h_i is the probability of a hit to level i and misses to all the previous levels $j < i$. Thus

$$h_i = H_i - H_{i-1} = F_{i-1} - F_i \qquad i=1,2,\ldots n$$

and

$$F_0 = 1 \ , \quad F_n = 0 \qquad \text{by convention.}$$

Therefore, the expression for T can be simplified to

$$\begin{aligned} T &= t_i + \sum_{i=2}^{n} F_{i-1} t_i \\ &= t_1 + \sum_{i=2}^{n} t_i F_{i-1}(K_{i-1}; s_{i-1}) \quad . \end{aligned}$$

The cost of the i^{th} memory level is modeled by a posynomial function of its capacity, that is

$$DEVCOST_i = \sum_j c_{ij} K_i^{\gamma_{ij}} \qquad c_{ij} \geq 0 \ , \ i=1,2,\ldots,n$$

The total system cost is then given by $\sum_{i=1}^{n} DEVCOST_i$.

The failure rate, λ_i, of memory level i is a posynomial function of its capacity, that is

$$\lambda_i = \sum_j d_{ij} K_i^{\beta_{ij}} \ , \quad d_{ij} > 0 \ , \quad i=1,2,\ldots n \quad .$$

Note that the military standard MIL-217B [9] suggests that λ_i is a *linear function of capacity* K_i. A failure in any level is assumed to cause the entire hierarchy to fail. Thus, from the reliability point of view, it is a series system. If we further assume that the failure rate of each level is independent of its age (or equivalently, the lifetimes are exponentially distributed), the hierarchy failure rate, λ, is also time-independent, and is given by [1],

$$\lambda = \sum_{i=1}^{n} \lambda_i \quad .$$

For such a system, minimizing the failure rate is equivalent to maximizing the reliability [1].

We are now ready to define the optimization problem for the design of the storage hierarchy. Assume that the hierarchy access time is not to exceed T_0 and the system cost is constrained by BUDGET.

$$\min \quad \lambda = \sum_{i=1}^{n} \sum_j d_{ij} K_i^{\beta_{ij}} \tag{1a}$$

such that

$$T = t_1 + \sum_{i=2}^{n} t_i F_{i-1}(K_{i-1}; s_{i-1}) \leq T_0 \qquad (1b)$$

$$\sum_{i=1}^{n} \sum_{j} c_{ij} K_i^{\gamma_{ij}} \leq \text{BUDGET} \qquad (1c)$$

$$K_i > 0 \ , \quad i=1,2,\ldots,n-1 \qquad (1d)$$

If we assume that the miss ratio function, F_i, is a posynomial function of capacity K_i (that is, $F_i = \sum_j a_{ij} K_i^{-\alpha_{ij}}$, $a_{ij} \geq 0$, $i=1,2,\ldots,n-1$), then the design problem (1) above is a standard geometric programming problem. From the theory of geometric programming [13], we get the following result:

<u>Theorem 1</u>:

Any relative minimum of the design problem (1) is also its global minimum.

The implication of this result is that any standard technique for nonlinear optimization will locate the global minimum of the design problem. Alternatively, geometric programming techniques can be used to solve the problem. In the next section, we consider a simplification of this problem so that a closed form solution can be obtained.

3. Simplified Design Problem

In the previous section, the miss ratio function, the device cost function, and the failure rate function were assumed to be quite general posynomial functions. In this section we make the following

simplifying assumptions (that is, restrict the number of terms in the posynomials):

$$F_i = a_i K_i^{-\alpha_i} \;, \quad i=1,2,\ldots,n-1 \qquad \text{(one term only)}$$

$$\lambda_i = d_{0i} + d_{1i} K_i^{\beta_i} \;, \quad i=1,2,\ldots,n \qquad \text{(two terms only)}$$

$$DEVCOST_i = c_{0i} + c_{1i} K_i^{\gamma_i} \;, \quad i=1,2,\ldots,n \qquad \text{(two terms only)}$$

Note that, since K_n is fixed, λ_n and $DEVCOST_n$ are fixed. Define

$$d_0 = \sum_{i=1}^{n} d_{0i} + d_{1n} K_n^{\beta_n} \;,$$

$$c_0 = \sum_{i=1}^{n} c_{0i} + c_{1n} K_n^{\gamma_n} \;.$$

Now the design problem (1) can be rewritten as

$$\text{min.} \quad d_0 + \sum_{i=1}^{n-1} d_{1i} K_i^{\beta_i} \tag{2a}$$

$$\text{s.t.} \quad t_1 + \sum_{i=2}^{n} t_i a_{i-1} K_{i-1}^{-\alpha_{i-1}} \leq T_0 \tag{2b}$$

$$\sum_{i=1}^{n-1} c_{1i} K_i^{\gamma_i} \leq BUDGET - c_0 \tag{2c}$$

$$K_i > 0 \;, \quad i=1,2,\ldots,n-1 \tag{2d}$$

With these assumptions, it can be shown that [11] the solution to design problem (1) can be decomposed into two steps. The first step is to obtain the solution to the following subproblem:

$$\text{min.} \quad d_0 + \sum_{i=1}^{n-1} d_{1i} K_i^{\beta_i} \tag{3a}$$

$$\text{s.t.} \quad t_1 + \sum_{i=2}^{n} t_i a_{i-1} K_{i-1}^{-\alpha_{i-1}} \leq T_0 \tag{3b}$$

$$K_i > 0 \ , \quad i-1,2,\ldots,n-1 \tag{3c}$$

In particular, we have removed the cost costraint. Assume that a solution to problem (3) is given by K_1, K_2, ..., K_{n-1}. Now if

$$\sum_{i=1}^{n-1} c_{1i} K_i^{\gamma_i} \leq \text{BUDGET} - c_0$$

then this is also the solution to the overall design problem (2). On the other hand, if

$$\sum_{i=1}^{n-1} c_{1i} K_i^{\gamma_i} > \text{BUDGET} - c_0$$

then the original design problem (2) possesses no feasible solution. In other words, the requirements of the design problem are incompatible. Due to this result, we will, henceforth, omit the cost constraint from the problem specification. We will assume that the requirements are compatible.

With the further restriction that $\alpha_i = \alpha$ and $\beta_i = \beta$ for all i, the following closed form solution to design problem (3) can be obtained [11].

$$K_i^* = \left[\frac{\sum_{i=1}^{n-1} \left[\frac{\beta d_{1i}}{\alpha}\right]^{\frac{\alpha}{\alpha+\beta}} (a_i t_{i+1})^{\frac{\beta}{\alpha+\beta}}}{T_0 - t_1} \right]^{1/\alpha} * \left[\frac{a_i t_{i+1} \alpha}{\beta d_{1i}} \right]^{\frac{1}{\alpha+\beta}} \tag{4}$$

As an example, we take the input parameters shown in Table 1 for the design of a 3-level storage hierarchy. The data on the miss ratio function is derived from [8]. All other input parameters are based on the data reported in [2,5]. The resulting optimal capacities are also shown in Table 1.

Next we consider the problem with the same parameters, except that the maximum allowable hierarchy access time, is to be varied. The result of such a sensitivity analysis is shown in Figure 2. We have plotted the optimal hierarchy MTTF (equal to the reciprocal of the system failure rate) as a function of the maximum allowable hierarchy access time T_0. In Figure 3, we have plotted the cost of the optimally designed hierarchy as a function of T_0.

4. Including the Selection of CPU Speed

The model of linear storage hierarchy discussed up to this point does not address the selection of CPU speed. We have also ignored the effect of CPU instruction execution delay from performance considerations. Let t_0 be the average instruction interpretation and execution time (i.e., $1/t_0$ is the instruction execution rate assuming an infinitely fast memory hierarchy). Let A be the average number of memory references per instruction. Gecsei and Lukes [7] estimate that $A \cong 2$ while Snow and Siewiorek [2] estimate that $A \cong 1.162$.

The average execution time per instruction is now given by

$$t_0 + A[t_1 + \sum_{i=2}^{n} t_i F(K_{i-1})]$$

Apart from the memory capacities $K_1, K_2, \ldots, K_{n-1}$, we consider t_0 also to be chosen by optimization. We assume that t_0 is a power function of the complexity, G_0, of the CPU:

$$t_0 = B_{1,0} G_0^{-\alpha_0} + B_{0,0}$$

G_0 may be equated to the gate count, or any related measure of complexity. The CPU failure rate, λ_0, is also a power function of the complexity:

$$\lambda_0 = d_{00} + d_{10} G_0^{\beta_0}$$

Finally, the CPU cost is modeled by

$$c_{00} + c_{10} G_0^{\gamma_0} \quad .$$

The design problem (1) can be redefined as follows:

$$\text{min.} \quad d_0 + \sum_{i=1}^{n-1} d_{1i} K_i^{\beta_i} + d_{10} G_0^{\beta_0} \tag{5a}$$

$$\text{s.t.} \quad B_{10} G_0^{-\alpha_0} + A[t_1 + \sum_{i=2}^{n} t_i F(K_{i-1})] \leq \frac{1}{\text{IER}} - B_{00} \tag{5b}$$

$$\text{and} \quad c_{00} + c_{10} G_0^{\gamma_0} + \sum_{i=1}^{n} (c_{0i} + c_{1i} K_i^{\gamma_i}) \leq \text{BUDGET} \tag{5c}$$

$$G_0 > 0 \tag{5d}$$

$$K_i > 0 \tag{5e}$$

As in section 3, we use the miss ratio function

$$F_i = a_i K_i^{-\alpha_i} \quad , \quad i=1,2,\ldots,n-1 \quad .$$

IER is the minimum required instruction execution rate for the system consisting of the CPU and the storage hierarchy. Also, $d_0 = \sum_{i=0}^{n} d_{0i} + d_{1n} K_n^{\beta_n}$. As before, we can solve the subproblem ignoring the cost constraint. Assuming that $\alpha_i = \alpha$, $\beta_i = \beta$ for $i=1,2,\ldots,n-1$, it can be shown that the CPU delay t_0 must satisfy the equation [11]:

$$\frac{1}{A*IER} - \frac{t_0}{A} - t_1 = t_0^{\rho} \left[\frac{\alpha_0}{A\beta_0 a_{10}}\right]^{\frac{\alpha}{\alpha+\beta}} \frac{\sum_{i=1}^{n-1} (a_i t_{i+1})^{\frac{\beta}{\alpha+\beta}} \left[\frac{a_{1i}\beta}{\alpha}\right]^{\frac{\alpha}{\alpha+\beta}}}{B_{1,0}^{\frac{\alpha\beta_0}{\alpha_0(\alpha+\beta)}}}$$

where

$$\rho = \frac{\alpha\ (\alpha_0+\beta)}{(\alpha+\beta)\alpha_0}$$

This is a nonlinear equation with one unknown and can be solved using the standard iterative techniques to obtain the optimal value of t_0. Now the optimal values of the memory capacity K_i $(i=1,2,\ldots,n-1)$ can be obtained from equation (4) after substituting

$$\frac{1}{A}\left(\frac{1}{IER} - At_1 - t_0\right)$$

for the term $T_0 - t_1$.

As an example, assume we want to design a system which is capable of executing at 0.2 MIPS. Other input parameters are shown in Table 2. The parameter values are based on data reported in [2,5,8]. The optimal solution is also shown in Table 2. The design roughly corresponds to a typical PDP-11/60 configuration [2].

Next, we perform sensitivity analysis varying the specified value of the IER. The system MTTF and the system cost are plotted as functions of IER in Figures 4 and 5, respectively.

5. Concluding Remarks

We have developed a reliability-oriented design model for linear storage hierarchies. The decision variables are the capacities

of each memory level and the speed of the CPU. The design problem is set up as a geometric programming problem with the objective of maximizing system reliability subject to a cost and a performance constraint. Any relative optimum is a globally optimum solution to the design problem. Closed form expressions for the device capacaties are obtained. Examples illustrating the usefulness of this model are also presented.

Acknowledgements

I would like to thank Jose de la Cruz for his programming support and J.B. Clary of the Research Triangle Institute for many technical discussions.

6. References

[1] R.E. Barlow and F. Proschan, Statistical Theory of Reliability and Life Testing, Holt, Rinehart and Winston, New York, 1975

[2] C.G. Bell, Mudge, and McNamara (eds.), Computer Engineering, Digital Press, 1978.

[3] C.K. Chow, "On Optimization of Memory Hierarchies", IBM J. Research and Development, vol. 18, May 1974, pp 194-203.

[4] C.K. Chow, "Determination of Cache's Capacity and its Matching Storage Hierarchy", IEEE Trans. Computers, vol. C-25, No. 2, February 1976, pp 157-164.

[5] J. Clary, A. Jai, S. Weikel, R. Salks, and D. Sieworek, "A Preliminary Study of Built-In-Test for the Military Computer Family", Technical Report, Research Triangle Institute, Research Triangle Park, N.C., August 1978.

[6] J. Gecsei, "Determining Hit Ratios for Multilevel Hierarchies", IBM J. of Research and Development, vol. 18, No. 4 (July 1974), pp 316-327.

[7] J. Gecsei and T.A. Lucas, "A Model for the Evaluation of Storage Hierarchies", IBM Systems Journal, vol. 13, No. 2 (1974), pp 163-178.

[8] J.E. MacDonald and K.L. Sigworth, "Storage Hierarchy Optimization Procedure", IBM J. of Research and Development, vol. 19, No. 2 (March 1975), pp 133-140.

[9] Military Standard MIL217b

[10] K.S. Trivedi, "Analytic Modeling of Computer Systems", IEEE Computer, vol. 11, No. 10 (October 1978), pp 38-55.

[11] K.S. Trivedi, article in preparation

[12] K.S. Trivedi and T.M. Sigmon, "Optimal Design of Linear Storage Hierarchies", submitted for publication, August 1979.

[13] C. Zener, Engineering Design by Geometric Programming, Wiley-Interscience, New York, 1971.

Table 1 : Example of a Linear Storage Hierarchy Design

INPUT PARAMETERS :

Number of Levels, $n = 3$

Desired Hierarchy Access Time, $T_0 = 2.41\ \mu sec$

Address Space Size, $K_n = 500K$ words

Miss Ratio Function, $F_i = a_i K_i^{-\alpha}$

where , $a_1 = 576$

, $a_2 = 7056$

, $\alpha = 1.449$

Memory Level	Access Time per word	Block Size (words)	t_i (μs)	d_{1i} per 10^6hr/word	c_{1i} \$/word
1 (cache)	0.1 μs	32	0.1	0.066	1.60
2 (main mem)	1 μs	128	32	0.004625	0.160
3 (back up) (memory)	10 μs	-	1280	-	0.016

$d_0 = 0$, $C_{0i} = 0$, $d_{0i} = 0$, $\beta_i = 1$, $\gamma_i = 1$ for all i .

OUTPUT (OPTIMAL DESIGN)

Memory Level	Optimal Capacity K_i (words)	Contribution to Access Time (μs)	Contribution to Failure Rate per 10^6hr	Contribution to cost (\$)
1	1.2K	0.1	79.2	1920
2	44.35K	0.641	204.7	7080
3	500K	1.669	-	8000
Totals		2.410	283.9	17,000

Table 2 : Example Design of a CPU and its Matching Storage Hierarchy

INPUT PARAMETERS :

Number of levels, $n = 3$
Desired Instruction Execution Rate, IER = 0.2 MIPS
Address Space Size, K_n = 500 K words

Miss Ratio Function, $F_i = a_i K_i^{-\alpha}$

where , $a_1 = 576$

, $a_2 = 7056$

, $\alpha = 1.449$

Number of Storage References per Instruction, A = 1.162

$B_{0,0} = 0$, $\alpha_0 = 1$, $B_{1,0} = 2.298 \times 10^4$ gate-μsec

$d_{00} = 0$, $\beta_0 = 1$, $d_{10} = 0.00988$ failures per gate per 10^6 hours

$c_{00} = \$20,000$, $\gamma_0 = 1$, $c_{10} = 0.61$ \$ per gate

All other parameters are equal to their corresponding values in Table 1.

OUTPUT (OPTIMAL DESIGN)

Memory Level	Delay, t_i (μs)	Complexity		Contribution to Failure Rate per 10^6 hr	Contribution to Cost (\$)
0 (CPU)	1.915	11971	Gate Count, G_0	118	27,300
1 (cache)	0.1	1.11K	Capac. (words)	72	1780
2 (main mem)	32	41.4K		191	6620
3 (backup memory)	1280	500K		-	8000
Totals				381	43,700

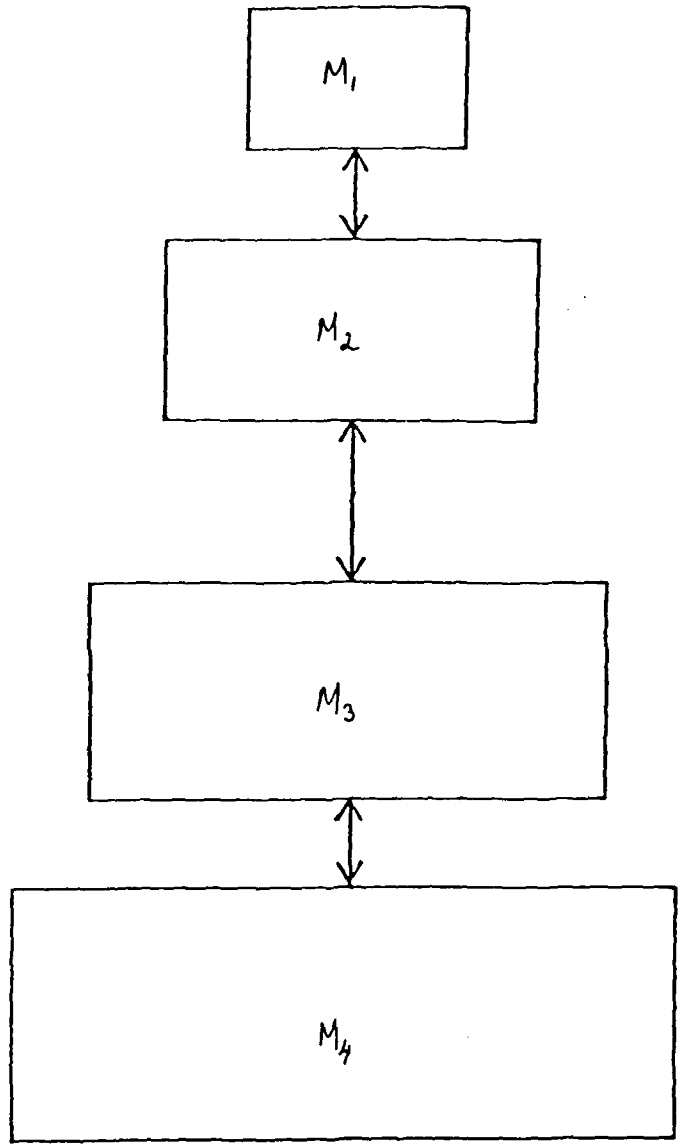

Figure 1. Linear Storage Hierarchy

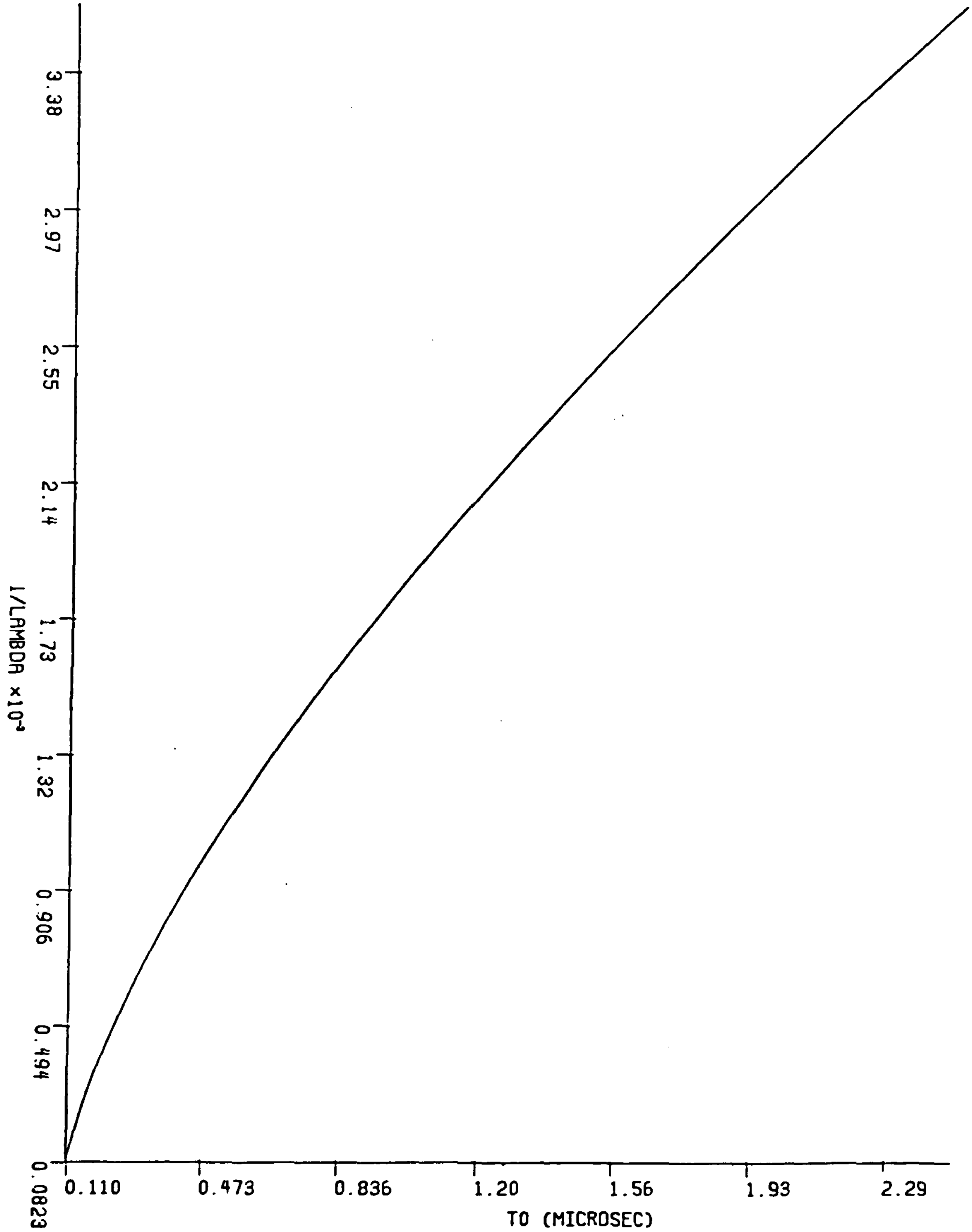

FIGURE 2. Hierarchy MTTF vs. Hierarchy Access Time

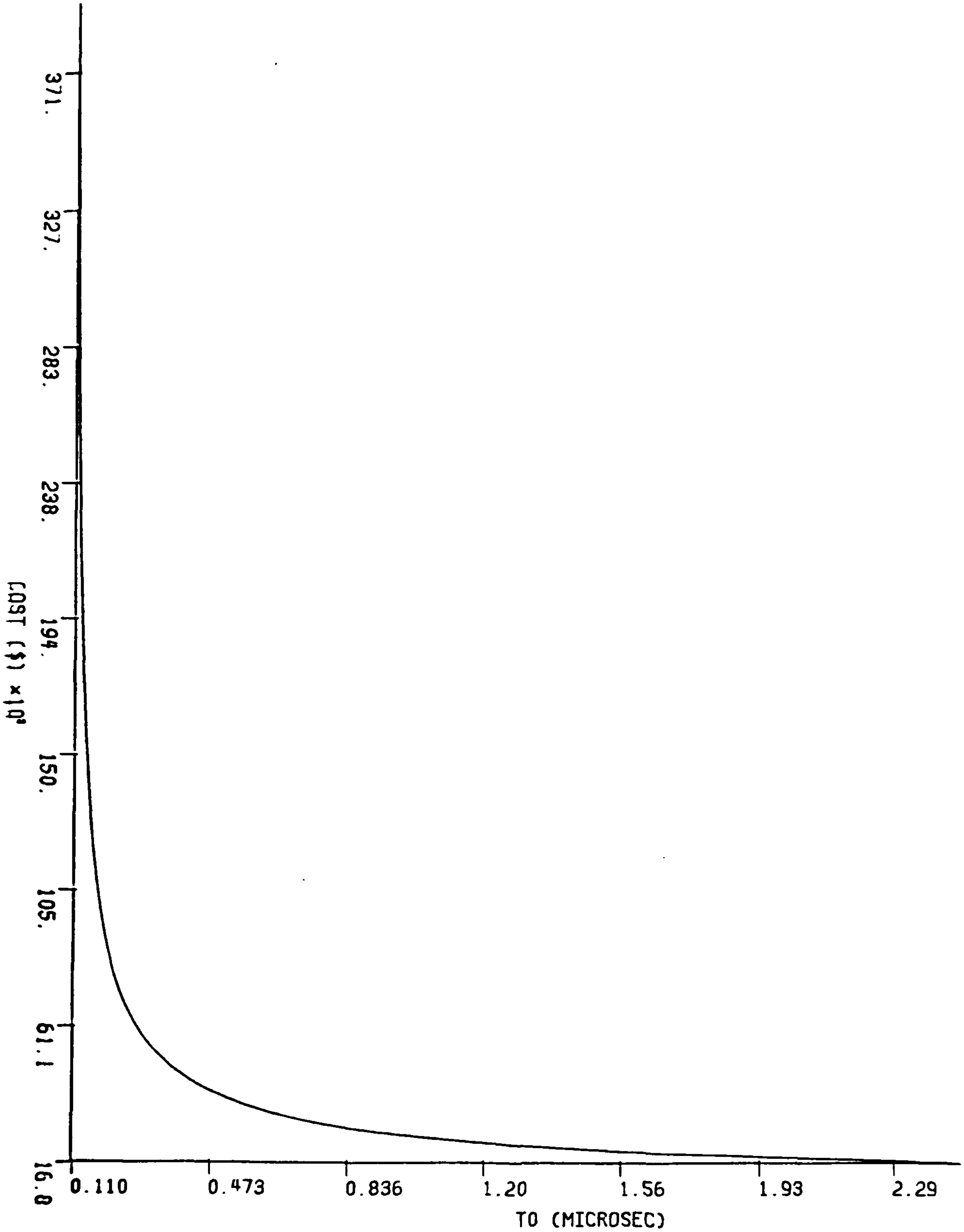

FIGURE 3. Hierarchy Cost vs. Hierarchy Access Time

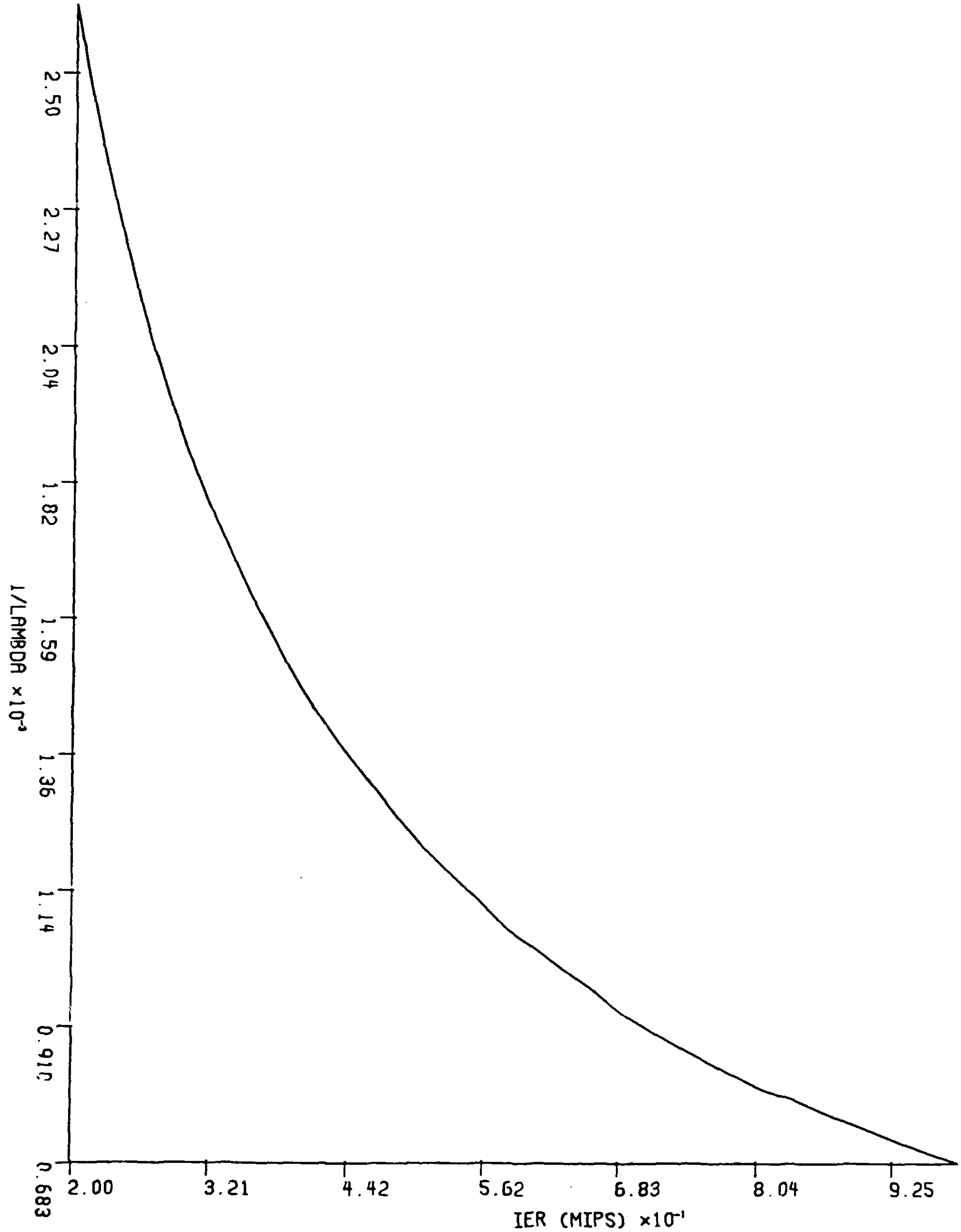

FIGURE 4. System MTTF vs. Instruction Execution Rate

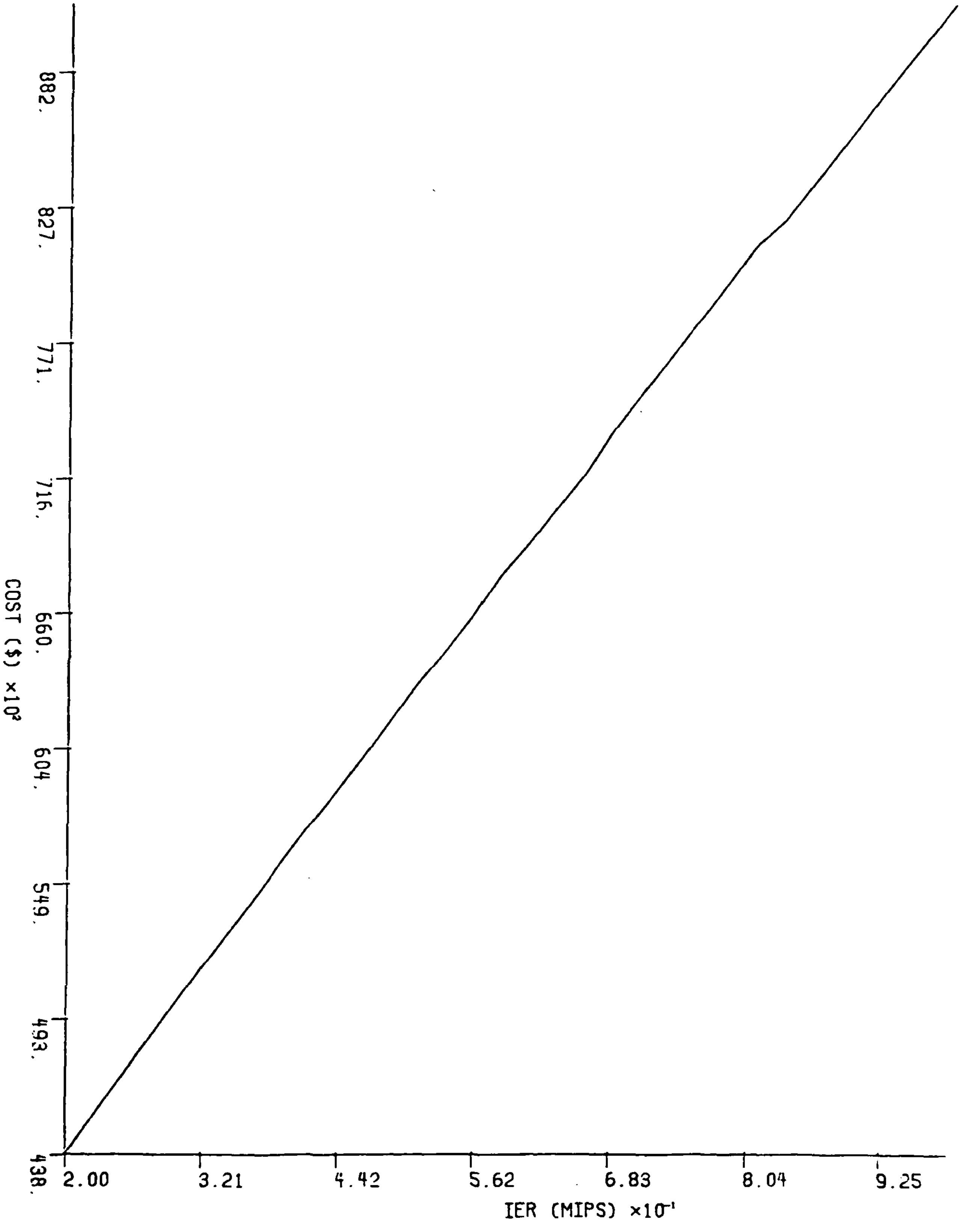

FIGURE 5. System Cost vs. Instruction Execution Rate